

COMPTE RENDUE DE TP N°1

I. Partie 1:

1. Pour se connecter en tant qu'administrateur à la base de données, on utilise la commande suivante :

```
SQL*Plus: Release 11.2.0.1.0 Production on Mer. Oct. 16 18:54:30 2019
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Entrez le nom utilisateur : connect/ as sysdba
Entrez le mot de passe :

Connect@ α :
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL>
```

2. Pour Afficher la structure de la table dba_users, on utilise la commande suivant :

```
SQL> desc dba_users
Nom                                NULL ?    Type
-----
USERNAME                           NOT NULL  VARCHAR2(30)
USER_ID                             NOT NULL  NUMBER
PASSWORD                           VARCHAR2(30)
ACCOUNT_STATUS                       NOT NULL  VARCHAR2(32)
LOCK_DATE                           DATE
EXPIRY_DATE                         DATE
DEFAULT_TABLESPACE                   NOT NULL  VARCHAR2(30)
TEMPORARY_TABLESPACE                 NOT NULL  VARCHAR2(30)
CREATED                             NOT NULL  DATE
PROFILE                             NOT NULL  VARCHAR2(30)
INITIAL_RSRC_CONSUMER_GROUP          VARCHAR2(30)
EXTERNAL_NAME                        VARCHAR2(4000)
PASSWORD_VERSIONS                    VARCHAR2(8)
EDITIONS_ENABLED                     VARCHAR2(1)
AUTHENTICATION_TYPE                  VARCHAR2(8)
```

3. Pour Afficher les noms de tous les utilisateurs on utilise la commande suivante :

```
SQL> select username from dba_users ;

USERNAME
-----
MGMT_VIEW
SYS
SYSTEM
DBSNMP
SYSMAN
HR
OUTLN
FLOWS_FILES
MDSYS
ORDSYS
EXFSYS
```

4. Tant qu'on est connecté qu'un administrateur, donc on a le droit pour faire des changements sur les autres utilisateurs en utilisant la commande suivante :

```
SQL> alter user hr identified by hr2019 account unlock ;  
  
Utilisateur modifié.
```

5. Pour connecter avec le login HR et le nouveau mot de passe on utilise la commande suivante :

```
SQL> connect hr/hr2019 ;  
Connecté.  
SQL>
```

Et pour vérifier l'utilisateur qu'est connecté à la base de données on utilise la commande suivante :

```
SQL> show user ;  
USER est "HR"  
SQL>
```

6. Pour afficher les tables d'un utilisateur, on doit d'abord connaître la structure d'une table en utilisant la commande suivante :

```
SQL> desc user_tables ;  
Nom                                NULL ?    Type  
-----  
TABLE_NAME                        NOT NULL  VARCHAR2(30)  
TABLESPACE_NAME                   VARCHAR2(30)  
CLUSTER_NAME                      VARCHAR2(30)  
IOT_NAME                         VARCHAR2(30)  
STATUS                           VARCHAR2(8)  
PCT_FREE                         NUMBER  
PCT_USED                         NUMBER  
INI_TRANS                        NUMBER  
MAX_TRANS                        NUMBER  
INITIAL_EXTENT                   NUMBER  
NEXT_EXTENT                      NUMBER  
MIN_EXTENTS                      NUMBER  
MAX_EXTENTS                      NUMBER  
PCT_INCREASE                     NUMBER  
FREELISTS                       NUMBER
```

Alors on déduit que chaque table est connue par son nom « Table_name », pour cela on utilise la commande suivante pour afficher tous les noms des tables qui appartiennent à l'utilisateur « HR » :

```
SQL> select table_name from user_tables ;  
  
TABLE_NAME  
-----  
REGIONS  
LOCATIONS  
DEPARTMENTS  
JOBS  
EMPLOYEES  
JOB_HISTORY  
COUNTRIES
```

7. Toujours en utilisant la commande « desc+ nom_table » pour afficher la structure d'une table de la base de données :

➤ **TABLE REGIONS**

```
SQL> desc REGIONS ;
Nom                                NULL ?   Type
-----
REGION_ID                         NOT NULL NUMBER
REGION_NAME                       VCHAR2(25)
```

➤ **TABLE LOCATIONS**

```
SQL> desc LOCATIONS ;
Nom                                NULL ?   Type
-----
LOCATION_ID                         NOT NULL NUMBER(4)
STREET_ADDRESS                     VCHAR2(40)
POSTAL_CODE                        VCHAR2(12)
CITY                               NOT NULL VCHAR2(30)
STATE_PROVINCE                     VCHAR2(25)
COUNTRY_ID                         CHAR(2)
```

➤ **TABLE DEPARTEMENTS**

```
SQL> desc DEPARTMENTS
Nom                                NULL ?   Type
-----
DEPARTMENT_ID                     NOT NULL NUMBER(4)
DEPARTMENT_NAME                    NOT NULL VCHAR2(30)
MANAGER_ID                         NUMBER(6)
LOCATION_ID                         NUMBER(4)
```

➤ **TABLE JOBS**

```
SQL> desc JOBS ;
Nom                                NULL ?   Type
-----
JOB_ID                             NOT NULL VCHAR2(10)
JOB_TITLE                           NOT NULL VCHAR2(35)
MIN_SALARY                          NUMBER(6)
MAX_SALARY                          NUMBER(6)
```

➤ **TABLE EMPLOYEES**

```
SQL> desc EMPLOYEES ;
Nom                                NULL ?   Type
-----
EMPLOYEE_ID                         NOT NULL NUMBER(6)
FIRST_NAME                          VCHAR2(20)
LAST_NAME                           NOT NULL VCHAR2(25)
EMAIL                               NOT NULL VCHAR2(25)
PHONE_NUMBER                        VCHAR2(20)
HIRE_DATE                           NOT NULL DATE
JOB_ID                              NOT NULL VCHAR2(10)
SALARY                              NUMBER(8,2)
COMMISSION_PCT                      NUMBER(2,2)
MANAGER_ID                          NUMBER(6)
DEPARTMENT_ID                       NUMBER(4)
```

➤ TABLE COUNTRIES

```
SQL> desc COUNTRIES ;
```

Nom	NULL ?	Type
COUNTRY_ID	NOT NULL	CHAR(2)
COUNTRY_NAME		VARCHAR2(40)
REGION_ID		NUMBER

8.

9. Pour Afficher le nombre de pays (countries) dans chaque region(region_name) on utilise la jointure car on utilise deux tables « countries et regions »:

```
SQL> select r.region_name , count(c.country_id) as nombre_country from regions r ,countries c
2  where r.region_id=c.region_id group by(region_name) ;
```

REGION_NAME	NOMBRE_COUNTRY
Middle East and Africa	6
Europe	8
Asia	6
Americas	5

10. Pour Afficher les numéros de départements ayant un directeur on utilise la commande suivante :

```
SQL> select department_id from departments where manager_id is not null ;
```

DEPARTMENT_ID
10
20
30
40
50
60
70
80
90
100
110

11 ligne(s) sélectionné(s).

11. Pour Afficher le nom et le prénom de l'employé ayant la fonction « président » on utilise la commande suivante :

```
SQL> select E.first_name , E.last_name from employees E ,jobs J
2  where J.job_id=E.job_id and J.job_title='President' ;
```

FIRST_NAME	LAST_NAME
Steven	King

12. - En utilisant la jointure entre la table « EMPLOYEES » et la table « DEPARTMENTS » pour afficher le nom du département affecté à l'employé « Steven King » :

```
SQL> select d.department_name from departments d ,employees e
  2  where d.department_id = e.department_id and e.first_name='Steven' and e.last_name='King' ;

DEPARTMENT_NAME
-----
Executive

SQL>
```

13. Pour Lister les départements qui se situent en Europe on utilise la jointure et les requêtes imbriquées comme suivante :

```
SQL> select d.department_name from departments d ,locations l ,countries c
  2  where d.location_id = l.location_id and l.country_id = c.country_id
  3  and c.region_id = (select region_id from regions where region_name = 'Europe') ;

DEPARTMENT_NAME
-----
Human Resources
Sales
Public Relations
```

14. Pour Lister les employés (last_name, first_name, revenu) qui ont un montant de revenu supérieur à 10000 (revenu=salaire+ salaire*commission_pct) on utilise la commande suivante :

```
SQL> select last_name ,first_name ,(salary+salary*commission_pct) as revunue
  2  from employees where(salary+salary*commission_pct > 10000 ) ;

LAST_NAME          FIRST_NAME          REVUNUE
-----
Russell            John                19600
Partners           Karen               17550
Errazuriz          Alberto             15600
Cambrault          Gerald              14300
Zlotkey            Eleni               12600
Tucker             Peter               13000
Bernstein          David               11875
Hall               Peter               11250
King               Janette             13500
Sully              Patrick             12825
McEwen             Allan               12150
```

15. Pour Afficher le nom complet du dernier employé embauché. On utilise la commande suivante :

```
SQL> select first_name ,last_name , hire_date from employees
  2  where hire_date = (select max(hire_date) from employees) ;

FIRST_NAME          LAST_NAME          HIRE_DAT
-----
Amit                Banda              21/04/08
Sundita             Kumar              21/04/08
```

II. Partie 2 :

1. Pour Se connecter en tant qu'administrateur on utilise la commande suivante :

```
Entrez le nom utilisateur : connect /as sysdba
Entrez le mot de passe :

Connect0 α :
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user ;
USER est "SYS"
```

2. On va créer un nouveau utilisateur dans notre base de donnée avec la commande suivant :

```
SQL> create user Rabie identified by 2000 default tablespace users temporary tablespace temp
2 profile default
3 account unlock ;

Utilisateur cr00.
```

3. Pour Autoriser la connexion et les ressources à l'utilisateur que nous avons crée on utilise la commande suivante :

```
SQL> GRANT connect, resource to Rabie ;

Autorisation de privil0ges (GRANT) accept0e.
```

4. Pour connecter en tant que nouveau utilisateur « Rabie » on utilise la commande suivante :

```
SQL> connect Rabie/2000 ;
Connect0.
SQL> show user ;
USER est "RABIE"
```

5. On Créer les tables du MLD :

- Table **SERVICES** :

```
SQL> create table services(id_service int not null ,nom_service varchar(20));

Table cr00e.
```

- Table **PIECE**:

```
SQL> create table piece(reference int not null ,
2 designation varchar(20) ,
3 prix float ,
4 poids float ,
5 couleur varchar(20) );

Table cr00e.
```

- Table **EMPLOYEE** :

```
SQL> create table employe(id_emp int not null ,
  2  nom varchar(20),
  3  prenom varchar(20),
  4  salaire float ,
  5  date_embauche date ,
  6  id_service int) ;
```

Table cr  e.

- Table **AFFECTATION**:

```
SQL> create table affectation(
  2  id_service int,
  3  reference int,
  4  date_affec date);
```

Table cr  e.

6. -On utilise les commandes suivantes pour ajouter les contraintes d'int  grit  s :

```
SQL> alter table services add constraint pk_service primary key(id_service);
```

Table modifi  e.

```
SQL> alter table piece add constraint pk_piece primary key(reference);
```

Table modifi  e.

```
SQL> alter table employe add constraint pk_employe primary key(id_emp);
```

Table modifi  e.

```
SQL> alter table employe add constraint fk_employe foreign key(id_service) references
  2  services(id_service)
  3  ;
```

Table modifi  e.

```
SQL> alter table affectation add constraint fk_affectation
  2  foreign key(id_service) references services(id_service);
```

Table modifi  e.

```
SQL> alter table affectation add constraint fk_affectatio foreign key(reference)
  2  references piece(reference) ;
```

Table modifi  e.

SQL>

7. Pour Ajouter une contrainte de domaine sur l'attribut COULEUR, qui ne doit prendre que les valeurs : ROUGE, VERTE, BLEUE et JAUNE on utilise la commande suivante :

```
SQL> alter table piece add constraint check_color CHECK(
  2  couleur='BLUE' or couleur='ROUGE' or couleur='VERTE' or couleur='JAUNE');

Table modifi e.
```

8. Pour Afficher les noms des contraintes avec l'Utilisation de table « user_constraints » on utilise la commande suivante :

```
SQL> select constraint_name from user_constraints ;

CONSTRAINT_NAME
-----
FK_EMPLOYE
FK_AFFECTATION
FK_AFFECTATIO
PK_EMPLOYE
PK_PIECE
PK_SERVICE
SYS_C0011097
SYS_C0011096
SYS_C0011098
CHECK_COLOR

10 ligne(s) s lectionn e(s).
```

9. Pour Ajouter l'attribut DIRECTEUR   la table SERVICES on utilise la commande suivante :

```
SQL> alter table services ADD(directeur varchar(20));

Table modifi e.
```